

A Data Model That Captures Clinical Reasoning About Patient Problems

Randolph C. Barrows Jr., MD, Stephen B. Johnson, PhD
Department of Medical Informatics
Columbia University, New York, NY

We describe a data model that has been implemented for the CPMC Ambulatory Care System, and exemplify its function for patient problems. The model captures some nuances of clinical thinking about patients that are not accommodated in most other models, such as an evolution of clinical understanding about patient problems. A record of this understanding has clinical utility, and serves research interests as well as medical audit concerns. The model is described with an example, and advantages and limitations in the current implementation are discussed.

INTRODUCTION

Cost-effective clinical computing will require systems capable of capturing the clinical encounter and providing introspective feedback to providers about diagnostic and treatment plans and other healthcare activities (or the lack of them) on their patients. A data schema that optimally models health care activities to facilitate these functional requirements is an area of ongoing research [1-6].

The Columbia-Presbyterian Medical Center (CPMC) is building an Ambulatory Care System (ACS) to support cost-effective clinical computing. The system utilizes CPMC resources that include a central patient database, a Medical Entities Dictionary (MED), a decision support system, and a Health Level 7 (HL7) "service access point" to move data between local systems and the central repository via HL7 messages [7].

The central patient data repository is a DB2 database that implements a generic relational model of medical activities with the advantage of comparatively few relations (i.e., tables in the database) [6]. Within this generic framework, we have implemented a data model to support ambulatory primary care activities with a complete computerized patient record (CPR). In addition to coded data capture and decision support activities, the model provides for the record of an evolution of medical insight regarding patients, an explicit representation of which is useful clinically, and for research, as well as necessary for medical audit purposes. This paper discusses some details of the model, focussing on implementation of the patient problem list.

BACKGROUND

All advanced CPR systems maintain a data dictionary to which patient observations are referenced, and accommodate some notion of time [8]. The HELP system, having an inpatient orientation, does not capture the notion of longitudinal patient problems [9]. In the STOR system, each problem is a "patient-item" with an onset and resolution time, and has an arbitrary number of associated time-specific "item-data" observations [10]. In the RMRS system, a problem list is a "multiple-choice multiple-answer" observation in the medical record file associated with a single date and time [11]. In TMR, problems may have an etiology link to other problems as well as dates of onset, resolution, and recurrence [12]. In the later three systems it is possible to infer the evolution of clinical reasoning about patient problems, but the flow is not explicitly represented. Enhancements to the public domain version of COSTAR V can capture some notion of clinical reasoning by allowing problems to be linked to other problems [13,14].

Rector uses a formalism of semantic networks to describe a comprehensive clinical data model that includes "meta-statements" about the decision making process and extensively captures clinical insight and accountability [1]. He makes a strong case for the model, which is being implemented in the PEN & PAD system. van Ginnekan discusses an "event-action" model which at a high level appears similar to ours; links between diagnosis-actions capture the evolution of clinical insight for a patient problem [4]. The model, however, is discussed as supportive of textual data entry rather than coded data; his notion of problem status pertains to certainty rather than disease activity; and the planned implementation is for the limited domain of cardiac failure rather than all of general medicine. Gouveia-Oliveira gives a complete conceptual data model centered on the idea of patient problems and their evolution [5]. The model is similar to ours in a capability to track the modification of a problem's identification or status and the modification of relationships between problems, but enforces problem-oriented progress notes while we have implemented encounter-based progress notes as being more supportive of the

actual clinical practice here at CPMC. Their system has yet to be implemented.

GENERIC RELATIONAL MODEL

A generic relational model of health care activities has the advantage of fewer relations (tables), which in practical implementation means fewer tables for which to maintain referential integrity, and easier database management [3,6]. Each generic relation has only a few generic attributes (columns). The model allows the degree of relations (number of attributes; columns in a table) to be abbreviated at the expense of the cardinality of the relations (number of rows). In effect, compared to more explicit relational models, the generic model transposes many data attributes so that attribute-value pairs are instead represented as tuples (rows) in a relation.

For example, instead of storing a CHEM-7 test battery with its 7 component tests as a single row of data (each test value in a separate column of the table), each component of the CHEM-7 is stored as a separate row consisting nominally of the attributes <component-test-id, value>. Another relation in the database preserves the notion of a CHEM-7 as the organizing entity for the component tests. A CHEM-7, however, is just one of many kinds of "services" that are performed for a patient, all of which can be represented in a single relation in the database. The data dictionary plays an integral role in the implementation of this data model by identifying what data are stored and the valid relationships between them [6].

In general, this modelling approach has proven to be practical, flexible, and efficient. All of the data captured during our ambulatory patient visits can be accommodated in only four base tables and their associated "components" tables. In terms of Entity-Relation modelling, the germane parts of the generic model are that patients may have one or more "services" performed for them, one or more "orders" made on them, one or more "assessments" made about them, and may participate in one or more general "management" events. With respect to ambulatory care, services performed for patients include blood pressure measurements, weight measurements, etc; orders include drug prescriptions, prescribed diets, etc; assessments include patient problems, allergies and adverse substance reactions, and compliance with preventive health recommendations; and management events include patient encounters such as an ambulatory visit with the associated provider record of that visit (i.e., the visit note). Other clinical applications, such as the CPMC Clinical Information System (CIS) and a resident sign-out application,

accommodate their data storage needs within the generic model as well.

PATIENT PROBLEM LIST

We have implemented a schema that supports the capture of coded and textual ambulatory patient data within the generic relational model described above. Since an accurate reflection of patient problems is quintessential to a good CPR for ambulatory care, we describe more explicitly how the problem list is modelled.

Within the ACS, users interact with a vocabulary server to search for, select and modify (coded) problems on their patients. The application context provides knowledge that the coded data are a "Patient Problem", which itself is just another coded concept in the MED. A conceptual semantic statement of this process is that "the provider has assessed that the patient has a problem". Likewise, a provider may assess through the user interface that the problem "cough" is now understood to be "chronic bronchitis", or that a formerly "active" problem now has an "inactive" status.

Thus, patient problems are dynamic; active problems become inactive or resolved, then flare and become active again. The nature of clinical thinking about problems often evolves over time. A problem initially described concretely as a symptom or finding will later be understood as some manifestation of a more abstract entity such as a disease or syndrome. Whether or not the patient's problem actually changed, clinical assessments of the problem do change from time to time. Each of these assessments captures a "state" of the medical understanding of the patient problem. Linking contiguous problem-states together over time can more accurately reflect the clinical thinking about a patient.

PROBLEM STATE MODEL

In our model, a problem is a linked thread of problem-state assessments about a patient. Each problem-state is characterized by a status (Active/Inactive/Resolved) and a value (e.g., code for Bronchitis), and may have a link to the problem-state from which it "evolved" (one type of defined link in our system). Each problem-state has a begin time and a (possibly open) end time of any appropriate degree of resolution. Figure 1 gives a simplified example how a patient problem is initially recorded. The top table is the assessment relation, indicating patient **abc** has been assessed to have a problem with status = A (Active) since time **t1**. The "key" attribute is actually an exact "observation time" of this assessment, which also serves as the

primary database key. This is a new problem, so the link field is null; i.e., the problem-state is not linked to any previous problem-state. Since the entry reflects the current state for this problem, the end time is open (null). Other attributes not shown record the identity of the provider making the assessment, for which organization, at which location, etc. Details of the problem state are indicated in the bottom table, which is a "components" relation. Here a component code identifies the datum, and the datum value may be some other dictionary code or a literal value. The relation indicates that **anemia** is the coded patient problem, and the provider has also entered a text comment about the problem which is associated with the current problem state.

key	pt_id	assess	status	link	begin	end
123	abc	<problem>	A		t1	

key	component	value
123	<problem>	<anemia>
123	<comment>	"pt says since childhood"

Figure 1. Top table contains an assessment event. Bottom table contains detail items for that event.

Figure 2 shows the relations when this problem "evolves" to a higher resolution of understanding, namely that this patient's **anemia** is actually **iron deficiency anemia**. The relations indicate that the tuple identified by **234** is the current state for this problem, now characterized by a status = A and value of (the code for) **iron deficiency anemia** since time **t2**. This problem-state is linked to the previous state for the problem, which is seen to have had a status = A and value of **anemia** from time **t1** to time **t2**.

key	pt_id	assess	status	link	begin	end
123	abc	<problem>	A		t1	t2
234	abc	<problem>	A	123	t2	

key	component	value
123	<problem>	<anemia>
123	<comment>	"pt says since childhood"
234	<problem>	<iron deficiency anemia>

Figure 2. Top table contains assessment events. Bottom table contains detail items.

Figure 3 shows the relations when this problem "evolves" to become inactive, for example, after

some period of appropriate iron replacement therapy. The relations indicate that the tuple identified by **345** is the current state for this problem, now characterized by status = I (Inactive) and value of (the code for) **iron deficiency anemia**. This problem-state is linked to the previous state for the problem, which is seen to have had a status = A and value of **iron deficiency anemia** from time **t2** to time **t3**. The later is itself linked to the original problem-state as before.

key	pt_id	assess	status	link	begin	end
123	abc	<problem>	A		t1	t2
234	abc	<problem>	A	123	t2	t3
345	abc	<problem>	I	234	t3	

key	component	value
123	<problem>	<anemia>
123	<comment>	"pt says since childhood"
234	<problem>	<iron deficiency anemia>
345	<problem>	<iron deficiency anemia>
345	<comment>	"ferritin nl after FeSO4"

Figure 3. Top table contains assessment events. Bottom table contains detail items.

Problems may be deleted, from the user's viewpoint, by "evolving" them to an "erased" (E) status, which is not retrieved into the user interface. Since users may retain problems on a "resolved" problem list, this is really only necessary to accommodate true data entry mistakes. The complete patient record, however, including all "erased" data, is always available to system administrators.

DISCUSSION

The above problem-state model of patient problems has been implemented for our ACS, which is in final testing stages and is nearing production use as of this writing (to be described in a future paper).

The current state of a problem, which is the state that should be placed on a patient's problem list in the user interface, is readily determined as those problem-states with an open end time. In our user interface, current problem-states are sorted according to their status and placed onto "active", "inactive", and "resolved" problem lists. Another possible display might place all current problem-states on a single problem list with their status indicated visually.

A user may select any problem from the problem list and review details, including an overview of its clinical evolution and when and by whom each problem-state assessment was made. This is useful

clinically to gain a better understanding of the evolution of disease in a patient and a fuller appreciation of the patient's current clinical state. Such an understanding would otherwise often require reading through old encounter notes, possibly searching years back in time.

In combination with meta-data from the data dictionary, this data model also provides for a class of research queries that can elucidate how the clinical understanding of patient problems evolves over time. For instance, "how often is symptomatic coronary artery disease first misdiagnosed as non-cardiac or atypical chest pain?". A definitive answer to such a query from other data models would require manual review of records, whereas the relationship is made explicit in this data model.

Note that the model accommodates divergence of patient problems that can occur as clinical understanding evolves. For example, an initial problem of **cough** might, after further study, lead to the two concurrent diagnoses of **asthma** and **post-nasal drip**. Similarly, the model can readily be extended to accommodate the convergence of problems that sometimes occurs as clinical understanding evolves when, for example, two problems such as **anemia** and **elevated LDH** are later understood to be manifestations of the single problem **hemolytic anemia**. The record can indicate that the later problem-state "evolved" from both previous problem-states by creating an additional link as a tuple in the components relation. Currently neither divergence nor convergence of patient problems is implemented for our system due to user interface logistics, although the semantics of the interaction have been worked out.

By a similar means of creating an arbitrary number of additional links via component tuples, observational data may be related as supportive of patient problem assessments. Again, this has not been implemented due to logistical details, but is recognized conceptually as an advantageous part of the model.

In summary, we have implemented a data model that regards each patient problem as a linked collection of (possibly one or more) clinically assessed problem-states. The model captures more nuances of clinical thinking than other well known CPR implementations. Analogous implementations for other types of ambulatory patient data, such as prescriptions and health maintenance activities, can be seen to have similar advantages.

References

1. Rector AL, Nowlan WA, Kay S, Goble CA,

Howkins TJ. A framework for modelling the electronic medical record. *Methods of Information in Medicine*. 1993; 32(2):109-19.

2. Buitendijk HJ (editor). HL7 patient care object model: draft 2. HL7 Patient Care SIG. 1993.

3. Dolin RH. A high-level object-oriented model for representing relationships in an electronic medical record. *Proceedings 18th SCAMC*. November 1994:514-8.

4. van Ginneken AM, Stam H, Duisterhout JS. A powerful macro-model for the computer patient record. *Proceedings 18th SCAMC*. November 1994: 496-500.

5. Gouveia-Oliveira A, Lopes Luisa. Formal representation of a conceptual model for the patient-based medical record. *Proceedings 17th SCAMC*. 1993:466-470.

6. Johnson S, Friedman C, Cimino JJ, Clark T, Hripcsak G, Clayton PD. Conceptual data model for a central patient database. *Proceedings 15th SCAMC*. 1991:381-5.

7. Clayton PD, Sideli RV, Sengupta S. Open architecture and integrated information at Columbia-Presbyterian Medical Center. *MD Computing*. 1992; 9(5):297-303.

8. McDonald CJ. Computer-stored medical record systems. *MD Computing*. 1988; 5(5):4-5.

9. Pryor TA. The HELP medical record system. *MD Computing*. 1988; 5(5):22-33.

10. Whiting-O'Keefe QE, Whiting A, Henke J. The STOR clinical information system. *MD Computing*. 1988; 5(5):8-21.

11. McDonald CJ, Blevins L, Tierney WM, Martin DK. The Regenstrief medical records. *MD Computing*. 1988; 5(5):34-47.

12. Stead WW, Hammond WE. Computer-based medical records: the centerpiece of TMR. *MD Computing*. 1988; 5(5):48-62.

13. Osburn AE, Neches NM, Shissler GE, Kittredge D. Enhancement to COSTAR with a problem oriented record structure and decision making support functions. *Proceedings 5th SCAMC*. 1981:124-127.

14. James Campbell, MD. University of Nebraska Medical Center: personal communication.